

회전을 고려한 판재 배치 문제의 유전 알고리즘 적용

Application of the Genetic Algorithm to the Layout Problem of the Pane Considering Rotation

이 금 탁, 김 훈 모
(Keum-Tark Lee and Hun-Mo Kim)

Abstract : A problem of relevant interest to some industries is that of the optimum two-dimensional layout. In this problem, one is given a number of rectangular sheets and an order for a specified number of each of certain types of two-dimensional regular and irregular shapes. The aim is to cut the shapes out of the sheets in such a way as to minimize the amount of waste produced. In this paper, we propose a genetic algorithms using rotation parameters by which the best pattern of layout is found.

Keywords : genetic algorithms, optimal two-dimensional layout, rotation parameter

I. 서론

직사각형 형태나 불규칙 패턴의 배치에 대한 문제는 2차원 형상들이 직사각형 같은 제한된 영역 안에서 불필요한 조각들이 가능한 한 최소가 되도록 혹은 배치된 개수가 최대가 되도록 알고리즘을 구현하는데 있다. 이러한 형상의 패턴 배치 문제는 생산성에 있어서 하나의 중요한 인자로 작용한다[1,3,4,9].

패턴배치는 크게 직사각형 패턴의 배치와 불규칙 패턴의 배치라는 두 가지로 나눌 수 있다. 직사각형의 배치에 관한 문제는 1961년의 Gilmore 등에 의해 논문이 발표되면서부터 본격적으로 시작되었는데 이들은 선형계획법(linear problem)과 낭색 함수(knapsack function)를 사용하여 1차원, 2차원 및 3차원 절단 문제를 풀었다[1,2]. Adamowicz 와 Albano는 2단계 접근법을 도입한 휴리스틱 알고리즘을 제안하였다. 이는 직사각형들을 스트립(strip)들로 형성시키는 단계와 동적계획법을 사용하여 스트립들의 최적배치에 들어가는 2단계로 구성되어 있다[5,6]. 그러나 이러한 경우 단순 직사각형 형태에 국한되어 있기 때문에 범용적이지 못한 문제점이 있다.

불규칙 패턴의 배치에 관한 연구는 Adamowicz 와 Albano에 의해 처음 발표되었다. 이 논문은 2단계 접근법에 관한 것으로 임의의 패턴 혹은 패턴들의 집단을 둘러싸는 최소크기 직사각형 모듈의 생성 단계와 이 모듈들의 최적 배치 단계를 통해 해를 구하였다. Albano는 휴리스틱 방법과 동적 계획법을 조합해 대략적인 배치를 자동으로 구한 다음 사용자에 의해 대화식으로 개선하는 방법을 제안하였다[6]. 그러나 이러한 휴리스틱 방법의 경우 지역해에 빠질 염려가 남아있다.

본 연구에서는 패턴 배치문제를 유전자 알고리즘을 이용하여 접근하고자 한다. 이 방법은 배치할 패턴에 순번을 부여하고 이 번호의 순서를 유전자 문자열로 사용하는 것

이다. 기존의 패턴순서열의 유전자 알고리즘에 관한 연구에서는 단순히 주어진 순서열에 의해 형상들을 배치하여 형상들 사이의 낭비공간이 많이 남아있는 단점이 있다[7,8]. 따라서 본 연구에서는 각각의 형상에 대한 일련의 순서열을 유전인자로 사용함과 동시에 각 형상들에 대한 회전을 유전인자로 고려하여 유전자 알고리즘을 수행함으로써 최적배치를 하는데 목적이 있다. 이는 각 형상에 대한 좌표를 유전인자로 할 경우 겹침 문제와 해에 접근하기 위한 시간이 많이 걸리는 단점을 극복하고 회전인자를 사용함으로써 낭비공간을 보다 줄일 수 있는 장점이 있다.

II. 알고리즘

유전자 알고리즘은 자연현상의 자연 도태와 유전의 메커니즘에 기반을 둔 확률적인 탐색알고리즘이다. 유전자 알고리즘은 해 자체가 아니라 해에 대한 집합의 코딩으로 작용하고 단일 해가 아닌 해들의 개체군으로부터 탐색하며 도함수들이나 다른 보조 지식이 아닌 결정론적 정보(적합도 함수)를 사용한다. 또한 유전자 알고리즘은 결정적인 규칙이 아닌 확률적인 친이규칙을 사용한다[10,11,12].

유전자 알고리즘을 적용하기 위해서 각각의 주어진 형상에 일련의 번호를 부여한 후 각 형상이 배치될 순서를 정한다. 이러한 순서는 유전자 알고리즘을 통해 구해진다. 먼저 유전자 알고리즘은 그림 1과 같이 1세대동안 선택, 교배, 돌연변이 연산을 수행한 후 형상들이 배치될 순서에 해당하는 순서열을 얻게된다. 이러한 배치 순서열을 배치 알고리즘에 넘겨주게 되며 순서열에 따라 배치 알고리즘을 적용한 후 면적을 계산하여 적합도 함수를 얻는다. 각각의 세대마다 이러한 과정을 반복하게 되면서 최적값을 찾아가 한다.

유전자 알고리즘을 적용하기 위해서 각각의 주어진 형상에 일련의 번호를 부여한 후 각 형상이 배치될 순서를 정한다. 이러한 순서는 유전자 알고리즘을 통해 구해진다. 먼저 유전자 알고리즘은 그림 1과 같이 1세대동안 선택, 교배, 돌연변이 연산을 수행한 후 형상들이 배치될 순서에 해당하는 순서열을 얻게된다. 이러한 배치 순서열을 배치 알고리즘에 넘겨주게 되며 순서열에 따라 배치 알고리즘

접수일자 : 1999. 12. 24., 수정완료 : 2000. 2. 24.

이금탁 : 성균관대학교 기계설계학과

김훈모 : 성균관대학교 기계공학부

※ 본 논문은 한국학술진흥재단 신진연구인력 연구 장려금 지원 하에 연구하였습니다.

을 적용한 후 면적을 계산하여 적합도 함수를 얻는다. 각각의 세대마다 이러한 과정을 반복하게 되면서 최적값을 찾고자 한다.

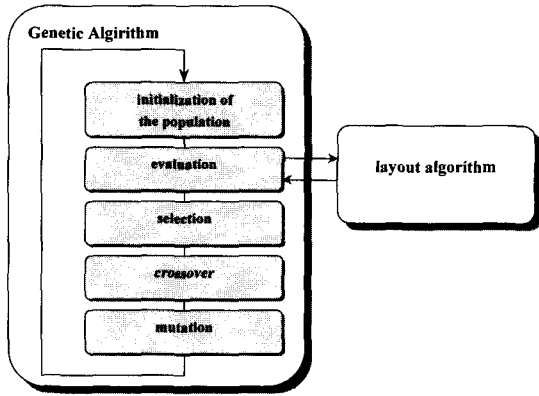


그림 1. 배치문제를 갖는 유전자 알고리즘.
Fig. 1. Genetic algorithm with layout problem.

본 연구에서는 두 가지의 알고리즘을 테스트하기 위해 유전 인자를 두 가지 형태로 사용하였다. 첫 번째는 각 형상들의 순서열만 적용한 경우와 두 번째는 형상들의 순서열과 각 형상들의 회전각을 동시에 적용한 두 가지 형태이다. 두 번째 형태는 그림 2와 같이 두 부분으로 나뉘어져 있다. 첫 번째 인자는 모든 부품에 대한 순서열을 나타내고 두 번째는 각각의 형상이 회전하는 각을 나타낸다. 회전각의 경우 8가지의 경우수로 했다. 즉 360도를 8가지의 각으로 나누어 각 형상이 회전할 수 있는 각도를 8가지로 제한하여 수행속도를 높이고자 하였다.

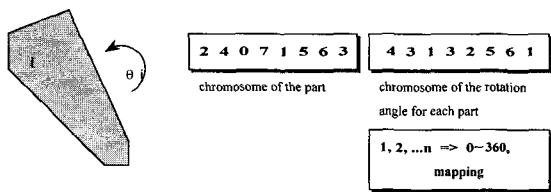


그림 2. 염색체.
Fig. 2. Chromosome.

낭비 면적의 최소화는 곧바로 적합도 함수로 이용할 수 있으며 (1)과 같이 계산한다.

$$\begin{aligned} \min S &= W \times L_{req} - \sum_{i=1}^N a_i & (1) \\ \text{subject to } & \sum_{i=1}^N \sum_{j=i+1}^N \beta_{i,j} = 0 \\ \beta_{i,j} &= \begin{cases} 0 & \text{Non-overlap} \\ 1 & \text{Overlap} \end{cases} \\ \text{Fitness} &= \frac{1}{S} \times a \end{aligned}$$

W는 배치될 판재의 고정된 높이이고 L_{req} 는 배치된 형

상을 모두 포함하는데 필요한 최소 폭을 나타낸다. a_i 는 임의의 형상 i 에 대한 면적을 나타내며 $\beta_{i,j}$ 는 형상 i 와 j 의 겹침여부를 나타내는 값이다. 적합도를 취하기 위해 먼저 필요한 면적에서 모든 형상 면적의 합 S 를 구한 후 이의 역수를 취한다. 적절한 스케일 값을 얻기 위해 적당한 상수값 a 를 곱하여 적합도값을 얻게 된다. 그림 3과 같은 다각형 형상의 면적은 (2)와 같이 구한다.

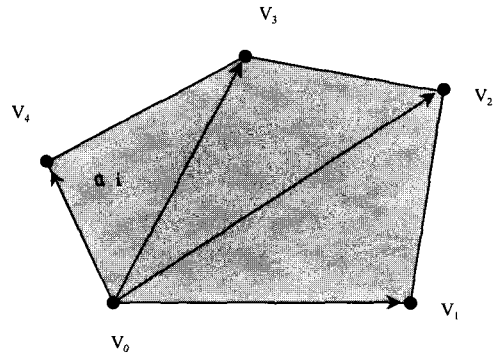


그림 3. 다각형 면적 계산.
Fig. 3. Calculation of the polygon.

$$a_i = \frac{1}{2} \left| \sum_{k=0}^{N-1} V_k \times V_{k-1} \right| \quad (2)$$

V_i 는 형상의 기준점 V_0 로부터 벡터를 나타낸다. 유전알고리즘에서 교배연산과 돌연변이 연산은 최적해에 수렴하는 속도뿐만 아니라 지역 최소해를 벗어나는 중요한 인자로 작용한다. 본 연구에서는 교배방법으로 Position-Based Crossover를 사용하였다. 이 방법은 첫 번째 부모유전자로부터 임의의 위치에 해당하는 인자를 선택한 후 이 선택된 인자들과 두 번째 부모 유전자에서 첫 번째 유전자로부터 선택된 인자에 없는 인자들을 추출해서 자손유전자를 만드는 방법이다. 또한 돌연변이 연산으로는 삼입돌연변이 방법을 취했다. 이 방법은 임의의 위치의 유전인자를 같은 염색체의 다른 부분의 위치에 있는 유전인자와 교환하는 방법이다.

그림 4는 순서열에 따라 y방향으로 배치한 결과의 가상 예로서 순서열과 회전을 동시에 유전인자로 적용하였을 경우 나타나게 될 배치결과를 보여준다.

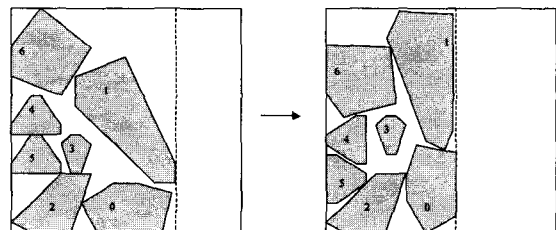


그림 4. 배치의 예.
Fig. 4. Example of the layout.

III. 배치문제

유전 알고리즘을 수행하여 얻은 순서열을 이용하여 배치 알고리즘 수행한다. 이 과정에서 모든 형상들은 겹치지 않게 y방향으로 이동하면서 배치되게 된다. 또한 판재의 최상 경계에 도달했을 경우 x방향으로 이동하면서 다시 최소 y점으로 이동해야 하며 같은 방법을 되풀이한다.

겹침 방지를 위한 알고리즘은 다음과 같이 수행된다. 두 형상에서 한 형상의 모든 꼭지점이 다른 형상에 포함되지 않는지를 조사하고 마찬가지로 다른 형상의 꼭지점이 또 다른 형상의 꼭지점에 포함하지 않는가를 조사하게 된다. 또한 꼭지점이 포함되지 않는 경우가 발생하는데 이때는 각각의 꼭지점을 연결하는 직선을 다른 형상의 직선과 교차하는지를 검사하게 된다.

모든 부품에 대해서 정해진 크기의 판재의 좌측 하단으로부터 하나의 부품을 y방향으로 계산된 값만큼 움직이면서 판재의 경계에 이르러 다시 x방향으로 이동 후 이 과정을 반복하게 되고 배치가 이루어지게 된다. 그림 5는 이러한 이동에 대한 예를 나타낸다. 이 과정에서 수행속도의 향상을 위해 y방향으로의 이동값 계산에 대한 알고리즘을 그림 6에서와 같이 수행하였다.

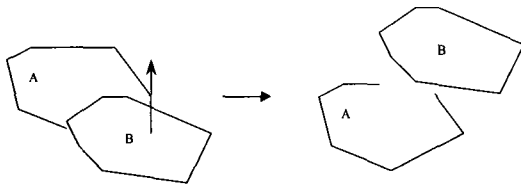


그림 5. 겹침에 대한 이동 문제.
Fig. 5. Move problem for overlapping.

먼저 두 형상 A, B가 x방향으로 겹치는 영역에서 최소 x좌표와 최대 x좌표 값을 각각 구한다. 즉 그림 6에서와 같이 형상 A와 겹치는 형상 B의 최소 x좌표를 지나는 수직선을 그었을 때 이 직선과 만나는 형상 A의 점을 P_1 , 형상 B와 만나는 점을 Q_1 이라 한다. 또한 형상 B와 겹치는 형상 A의 최대 x좌표를 지나는 수직선을 그었을 때 형상 A, 형상 B와 만나는 점을 각각 P_2, Q_2 라 한다. 그 다음 이동형상 B의 y방향으로의 최소점 Q_{min} 을 구하고 Q_1 과 Q_2 사이에 있는 점들 중에서 최소점 Q_{min} 의 x좌표를 지나는 수직선이 고정형상 A와 만나는 점 P_1 를 구한다. 이때 $P_1 - Q_{min} = d_1$ 이라 하자. P_1 과 P_2 사이에 있는 점들 중 최대점 P_{max} 를 구하고 P_{max} 의 x좌표를 지나는 수직선이 이동형상 B와 만나는 점을 구하고 이를 Q_1 라 한다.

Q_1 가 P_{max} 보다 크다면 이동형상 B를 이동할 필요가 없으므로 $d_2 = 0$ 으로 하고 Q_1 가 P_{max} 보다 작다면 $P_{max} - Q_1 = d_2$ 로 한다. 따라서 전체 이동거리는 (3)과 같다.

$$d = d_1 + d_2 \tag{3}$$

$$d_1 = P_1 - Q_{min}$$

$$d_2 = \begin{cases} 0, & \text{if } Q_1 \geq P_{max} \\ P_{max} - Q_1, & \text{if } Q_1 < P_{max} \end{cases}$$

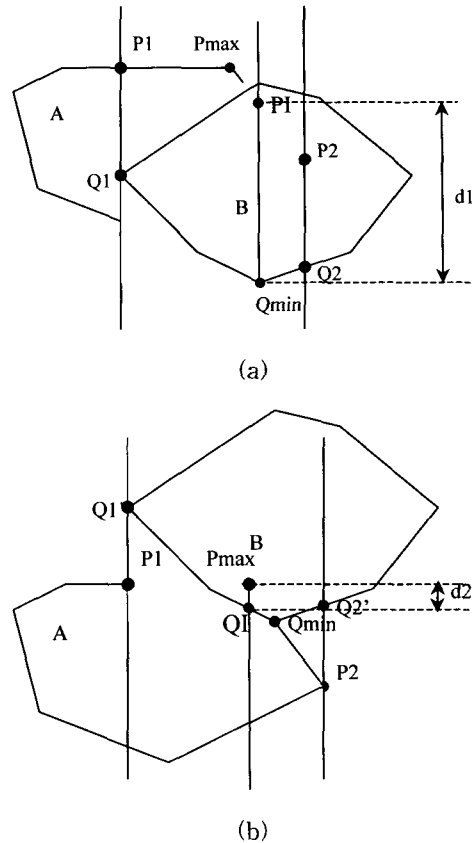


그림 6. 이동 거리의 계산.
Fig. 6. Calculation of move distance.

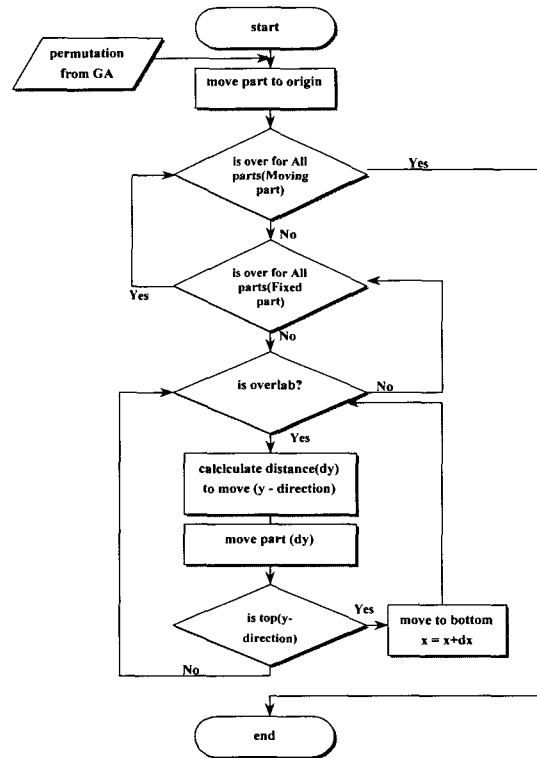


그림 7. 패턴 배치의 순서도.
Fig. 7. Flow chart of the pattern layout.

Q_i 가 P_{max} 보다 클 경우 한가지 문제가 발생할 수 있다. 즉 이동형상 B 가 Q_i 부분에서 오목한 형상일 경우 겹칠수가 있기 때문이다. 이 경우 본 연구에서는 dy 만큼의 이동거리를 두어 겹침을 방지하도록 하였다. 즉 이 경우의 실질적인 이동거리는 d_i+d_y+dy 가 된다.

전체적인 패턴 배치 알고리즘의 순서도와 겹침방지 알고리즘에 대한 프로그램 순서도가 그림 7, 그림 8에 나타나 있다.

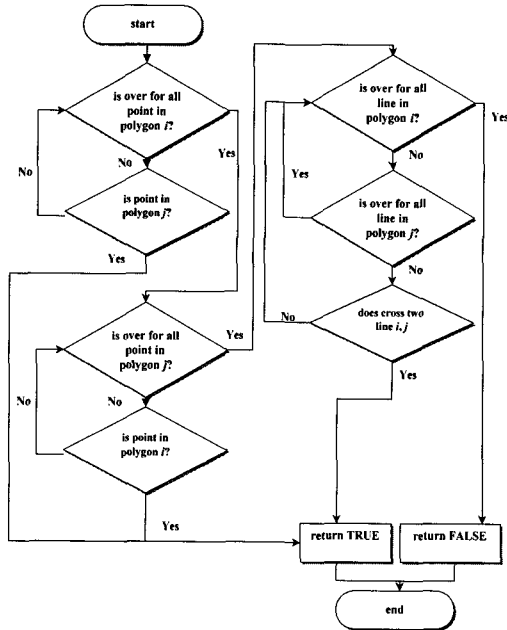
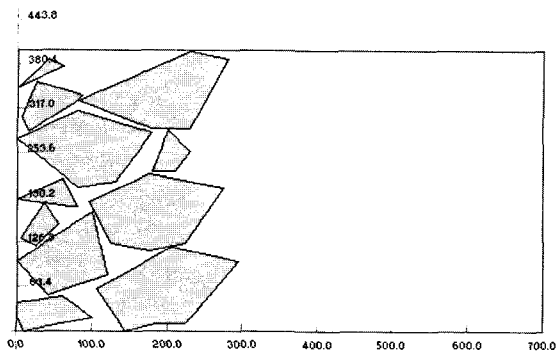


그림 8. 겹침 조사 알고리즘 순서도.
Fig. 8. Flow chart of overlap checking algorithm.

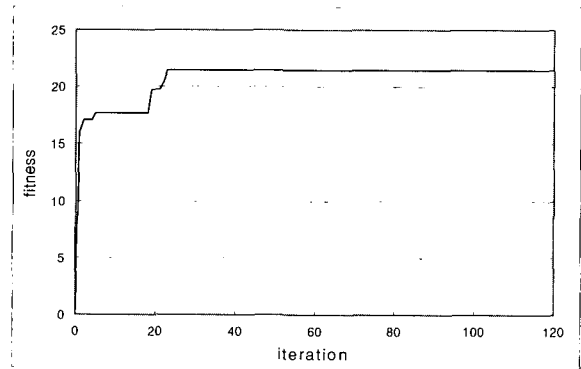
IV. 결과

본 연구에서는 회전 인자를 고려한 경우와 고려하지 않은 결과에 대한 비교를 수행하였으며 회전을 고려한 경우에서도 개체군수를 변화시켜가며 성능을 비교하였다.

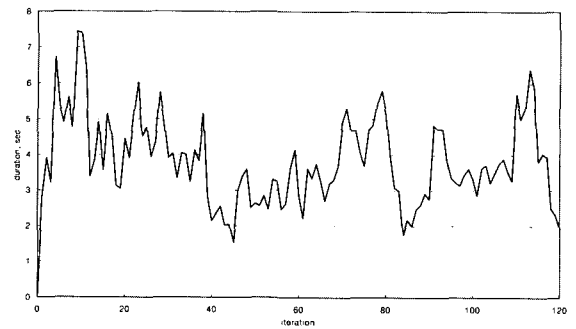
그림 9과 그림 10는 회전 인자를 고려하지 않은 경우와 회전인자를 고려한 유전자 알고리즘의 최적배치에 대한 시뮬레이션 결과를 보여준다. 이는 그림 9의 (a)와 그림 10의 (a)를 보면 낭비면적의 차이를 가시적으로 보여주며 회전인자를 포함한 수행결과가 보다 나음을 알 수 있다. 회



(a)



(b)



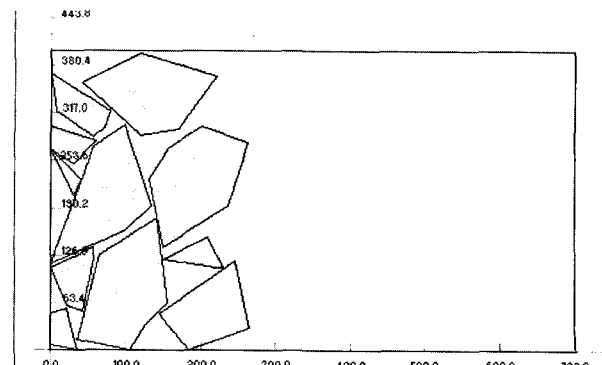
(c)

그림 9. (a) 회전 인자를 고려하지 않은 유전알고리즘의 수행 결과 (100세대에서의 적합도 : 21.4592), (b) 적합도, (c) 평균 수행시간(3.83 sec/gen.).

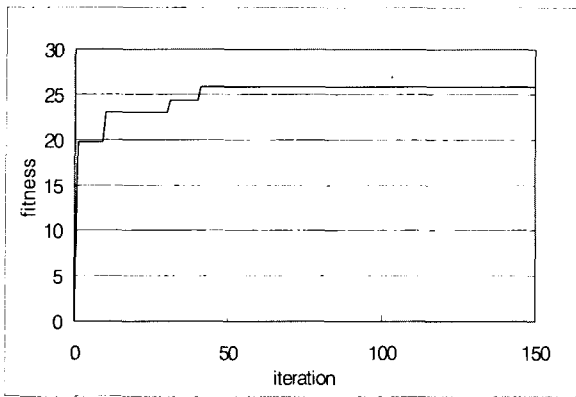
Fig. 9. (a) The result of the simulation of the GA without rotation parameter.(fitness at 100 generation : 21.4592), (b) Fitness, (c) Average duration(3.83 sec/gen.).

전 인자를 염색체에 포함하여 적용한 경우는 비회전의 경우에 발생할 수 있는 형상들간의 빈 공간영역을 줄일 수 있는 확률이 보다 커지므로 위와 같은 효과를 얻을 수 있었다. 그림 9의 (c)와 그림 10의 (c)는 두 경우에 대한 평균 수행시간을 나타낸 그림이다.

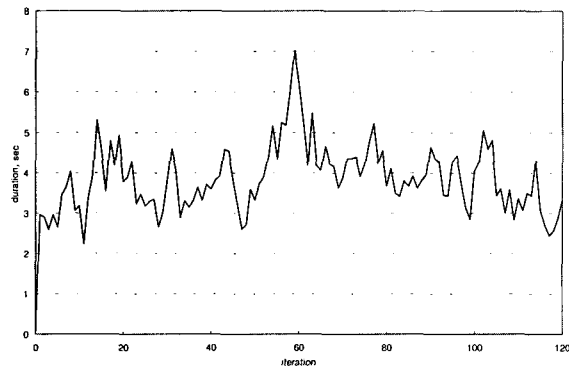
표 1은 두 결과에 대한 수치값으로 적합도에서 회전을



(a)



(b)



(c)

그림 10. (a) 회전인자를 고려한 유전알고리즘의 수행결과(개체군수 : 15, 100세대에서의 적합도 : 25.9067), (b) 적합도, (c) 평균 수행 시간(3.89 sec/gen.).

Fig. 10. (a) The result of the simulation of the GA with rotation parameter.(Number of population : 15, fitness at 100 generation : 25.9067), (b) Fitness, (c) Average duration (3.89 sec/gen.).

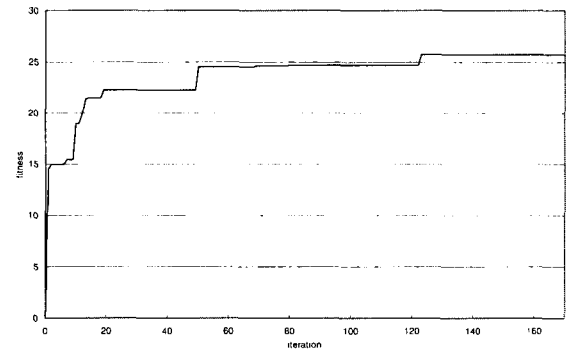
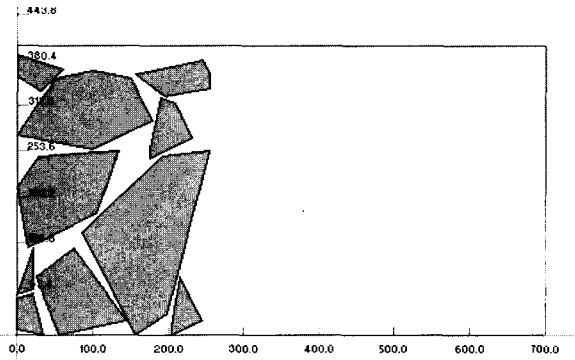
표 1. 회전 인자에 따른 비교.

Table 1. Comparison of the two case by the rotation parameter.

| | 비회전 | 회전 |
|--------|--------------|--------------|
| 다각형수 | 11 | 11 |
| 개체군수 | 15 | 15 |
| 적합도 | 21.4592 | 25.9067 |
| 교배 | 4쌍 | 4쌍 |
| 돌연변이율 | 0.05 | 0.05 |
| 평균수행시간 | 3.83sec/gen. | 3.89sec/gen. |

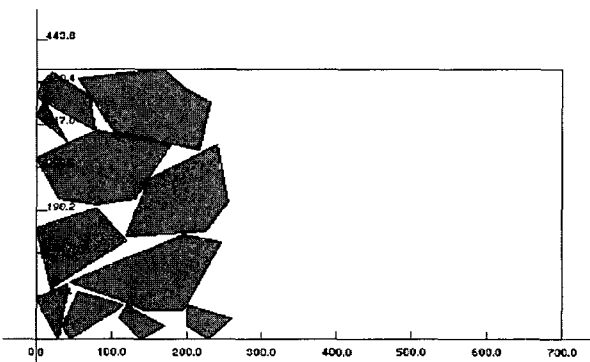
고려한 경우가 나음을 알수 있으며 수행 시간에 있어서는 비슷한 수치값을 보여준다. 이는 회전인자의 적용여부가 성능향상을 꾀하면서도 수행 시간에 그다지 큰 영향을 미치지 않음을 나타낸다.

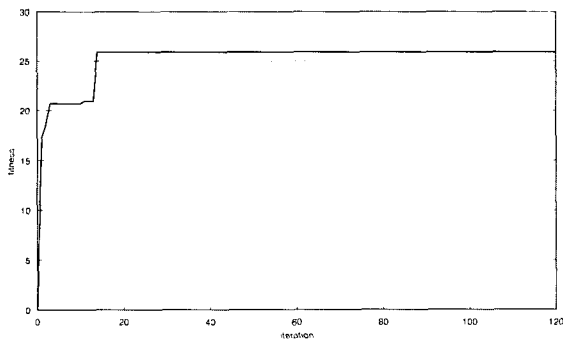
그림 11은 개체군의 수를 변화해 가며 수행한 결과를 나타낸다. 그림 11의 (a), (b), (c)에서처럼 개체군수를 6, 25, 35개로 변화해 가며 시물레이션을 수행한 결과를 보면 최적해에 접근하는 속도에서 개체수가 많은 경우 보다 빠름을 알 수 있다. 그러나 그림 12의 (a), (b), (c)에서 보는 바와 같이 평균 수행 시간은 상대적으로 떨어짐을 알 수 있다. 표 2는 이러한 적합도 값이나 평균 수행 시간을 수치적으로 보여준다. 개체군수를 바꾸면서 반복 테스트한 결과 개체군수는 대략 10~20개 정도에서 수행시간과 최적해에 접근하는 속도 두 가지 측면에서 만족시킴을 알 수 있었다.



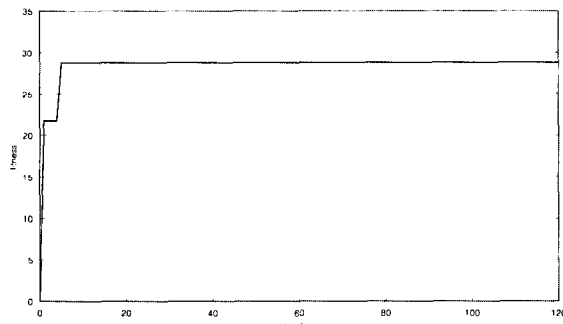
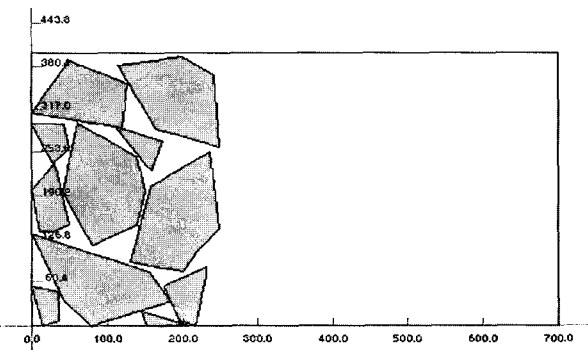
(a) 적합도(개체군 수 : 6, 100세대에서의 적합도 : 24.7200)

(a) The fitness of the GA(Number of population : 6, fitness at 100 generation : 24.7200)





- (b) 적합도(개체군 수 : 25, 100세대에서의 적합도 : 25.9067)
- (b) The fitness of the GA(Number of population : 25, fitness at 100 generation : 25.9067)

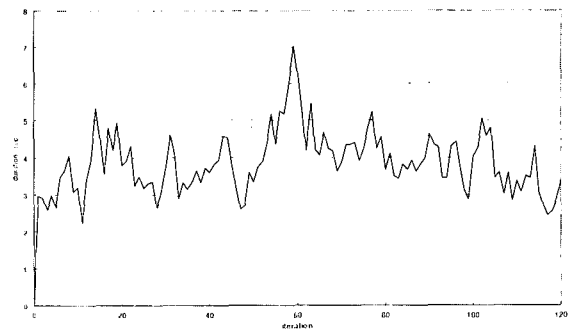


- (c) 적합도(개체군 수 : 35, 100세대에서의 적합도 : 28.8324)
- (c) The fitness of the GA(Number of population : 35, fitness at 100 generation : 28.8324)

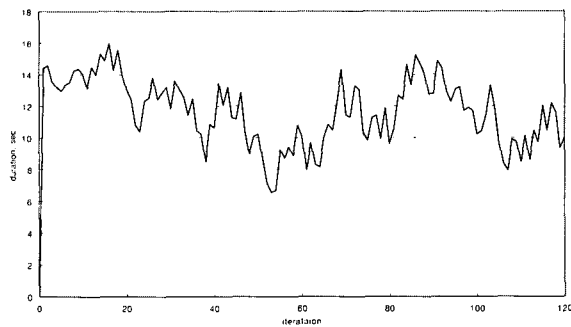
그림 11. 개체군수에 따른 적합도.
Fig. 11. The Fitness about number of population.

표 2. 개체군 수에 따른 비교.
Table. 2. Comparison of the three case by the number of population.

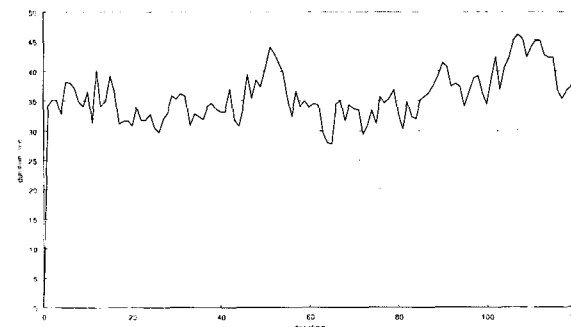
| | 개체군수 6 | 개체군수 25 | 개체군수 35 |
|---------|---------------|----------------|----------------|
| 다각형수 | 10 | 10 | 10 |
| 적합도 | 24.7200 | 25.9067 | 28.8324 |
| 교배 | 4쌍 | 4쌍 | 4쌍 |
| 돌연변이율 | 0.05 | 0.05 | 0.05 |
| 평균수행 시간 | 3.83 sec/gen. | 11.66 sec/gen. | 35.87 sec/gen. |



- (a) 수행 시간(개체군 수 : 6, 총 수행시간 : 459.6 sec, 평균 수행시간 : 3.83 sec/gen.)
- (a) GA duration per generation(population size : 6, Total duration (to 120 generation) : 459.6 sec, Average duration : 3.83 sec/gen.)



- (b) 수행 시간(개체군 수 : 20, 총 수행시간 : 1399.2 sec, 평균 수행시간 : 11.66 sec/gen.)
- (b) GA duration per generation(population size : 20, Total duration (to 120 generation) : 1399.2 sec, Average duration : 11.66 sec/gen)



- (c) 수행 시간(개체군 수 : 30, 총 수행시간 : 4204.4 sec, 평균 수행시간 : 35.87 sec/gen.)
- (c) GA duration per generation(population size : 30, Total duration (to 120 generation) : 4304.4 sec, Average duration : 35.87 sec/gen)

그림 12. 개체군 수에 따른 세대별 수행시간.
Fig. 12. Duration per generation about number of population.

그림 13은 본 시뮬레이션을 수행하기 위해 개발한 유전 알고리즘을 내장한 자동배치 프로그램으로서 회전인자, 개체군수 등의 인자를 변환하면서 수행할 수 있도록 하였다.

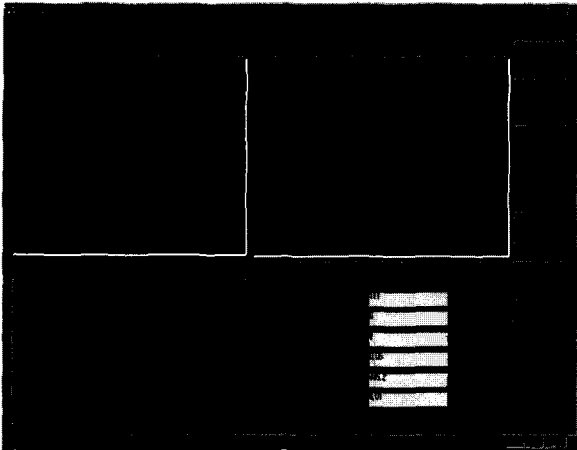


그림 13. 유전자 알고리즘을 이용한 자동배치 프로그램.

Fig. 13. Auto nesting program with genetic algorithm.

V. 결론

본 연구에서는 유전자 알고리즘을 이용한 형상들의 순서열 결정과 이를 이용한 배치를 수행하였다. 판재의 최적 배치에 대해서 회전인자에 회전각을 고려하여 두 가지 경우에 대한 시뮬레이션을 비교해보았다. 패턴순서만 정한 경우보다 회전인자를 고려한 배치알고리즘의 성능이 보다 나음을 알 수 있었고 회전인자를 고려한 경우에서 개체군수를 변화해 가며 시뮬레이션 한 결과 개체수가 많아질수록 최적해 수렴속도는 빨라지나 수행속도가 느려지는 단점이 있었다. 이 경우 수렴속도와 수행시간 모두 만족하기 위해서 적절한 개체군수의 선정이 필요하다. 수행 시간에 따른 결과를 보면 회전 인자를 고려한 경우와 고려하지 않은 경우에 있어서 수행 시간은 거의 차이가 없었고 다만 개체군수의 증가에 따른 수행시간이 급격히 길어진다. 회전인자를 고려해도 수행시간이 떨어지지 않음을 알 수 있었다.

현재 패턴순서를 입력으로 하는 형상 배치 알고리즘에서 어느 정도 시간지연이 일어난다. x방향으로는 단순히 일정한 값만큼 이동하도록 적용하고, y방향에 대해서만 배

치 알고리즘을 적용한데에 기인한 것으로 x방향에 대한 배치알고리즘을 추가하여 향후에 적용한다면 보다 나은 성능을 얻을 수 있을 것이다.

참고문헌

- [1] P. C. Gilmore and R. E. Gemory, "Multistage cutting stock problems of two and more dimensions", *Oper. Res.* vol. 13. pp. 94-120, 1965.
- [2] P. C. Gilmore and R. E. Gemory, "The theory and computation of knapsack functions", *Oper. Res.* vol. 14. pp. 1045-1074, 1966.
- [3] M. Adamowicz and A. Albano, "Nesting two-dimensional shapes in rectangular modules", *Computer-Aided Design*, vol. 8, no. 2, pp. 27-33, 1976.
- [4] S. C. Sarin, "Two-Dimensional Stock Cutting Problems and Solution Methodologies", *ASME, Journal of Engineering for Industry*, vol. 105, pp. 155-160, 1983.
- [5] Antonio Albano, "A method to improve two-dimensional layout", *Computer-Aided Design*, vol. 9, no. 1, pp. 48-52, 1977.
- [6] Antonio Albano and Giuseppe Sapuppo, "Optimal allocation of two-dimensional irregular shapes using heuristic search method", *IEEE Transactions on System, Man and Cybernetics*, pp. 242-248, 1980.
- [7] Gen Watanabe, and Tosihiko Ono, Determination of cutting layout of twodimensional pattern by genetic algorithm, 1997.
- [8] C. Bounsaythip, and S. Maouche, "A genetic approach to a nesting problem", *Proceedings of the Second Nordic Workshop on Genetic Algorithms and their Applications(2NWGA)*, pp. 89-104, 1996.
- [9] Lamousin, Waggenspack, and Dobson, "Nesting of complex 2-d parts within irregular boundaries", *ASME, Journal of Manufacturing Science and Engineering*, vol 118, pp. 615-622, 1996.
- [10] Mitsuo Gen and Runwei Cheng, *Genetic Algorithms & Engineering Design*, Wiley, pp. 1-41, 1997.
- [11] Lawrence Davis, *Handbook of Genetic Algorithms*, pp. 1-53. 1996.
- [12] Zbigniew Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd Edition, pp. 93-184, 1992.



이 금 탁

1997년 성균관대 기계설계학과 졸업. 동대학원 석사(1999), 관심분야는 분산 제어 및 CAD/CAM.

김 훈 모

제어·자동화·시스템공학 논문지 제4권, 제3호, 참조.